

## **MÓDULO CONVERSOR A/D DE 10 BITS DO dsPIC30F4011**

- ❑ O conversor Analógico – Digital (ADC) de alta velocidade permite conversões de um sinal de entrada analógico a um número digital de 10 bits. Este módulo é baseado na arquitetura de Registrador de Aproximações sucessivas (Successive Approximation Register – SAR) e fornece uma relação máxima de amostragem de 1 Msps.
- ❑ O módulo ADC tem 16 entradas analógicas as quais são multiplexadas em quatro amplificadores de amostragem e manutenção (sample and hold – S/H). A saída do sample and hold é a entrada ao conversor o qual gera o resultado.
- ❑ As voltagens de referência analógicas são selecionáveis por software, a tensão de alimentação do dispositivo (AVDD/AVSS) ou o nível de tensão sob os pinos (VREF+/VREF-). O módulo ADC tem a característica única de ser capaz de operar enquanto o dispositivo está em modo Sleep.

### **REGISTRADORES DE CONTROLE**

- ❑ O módulo ADC tem seis registradores de controle e status de 16 bits. Estes registradores são:
  - ADCON1: A/D Control Register 1
  - ADCON2: A/D Control Register 2
  - ADCON3: A/D Control Register 3
  - ADCHS: A/D Input Channel Select Register
  - ADPCFG: A/D Port Configuration Register
  - ADCSSL: A/D Input Scan Select Register

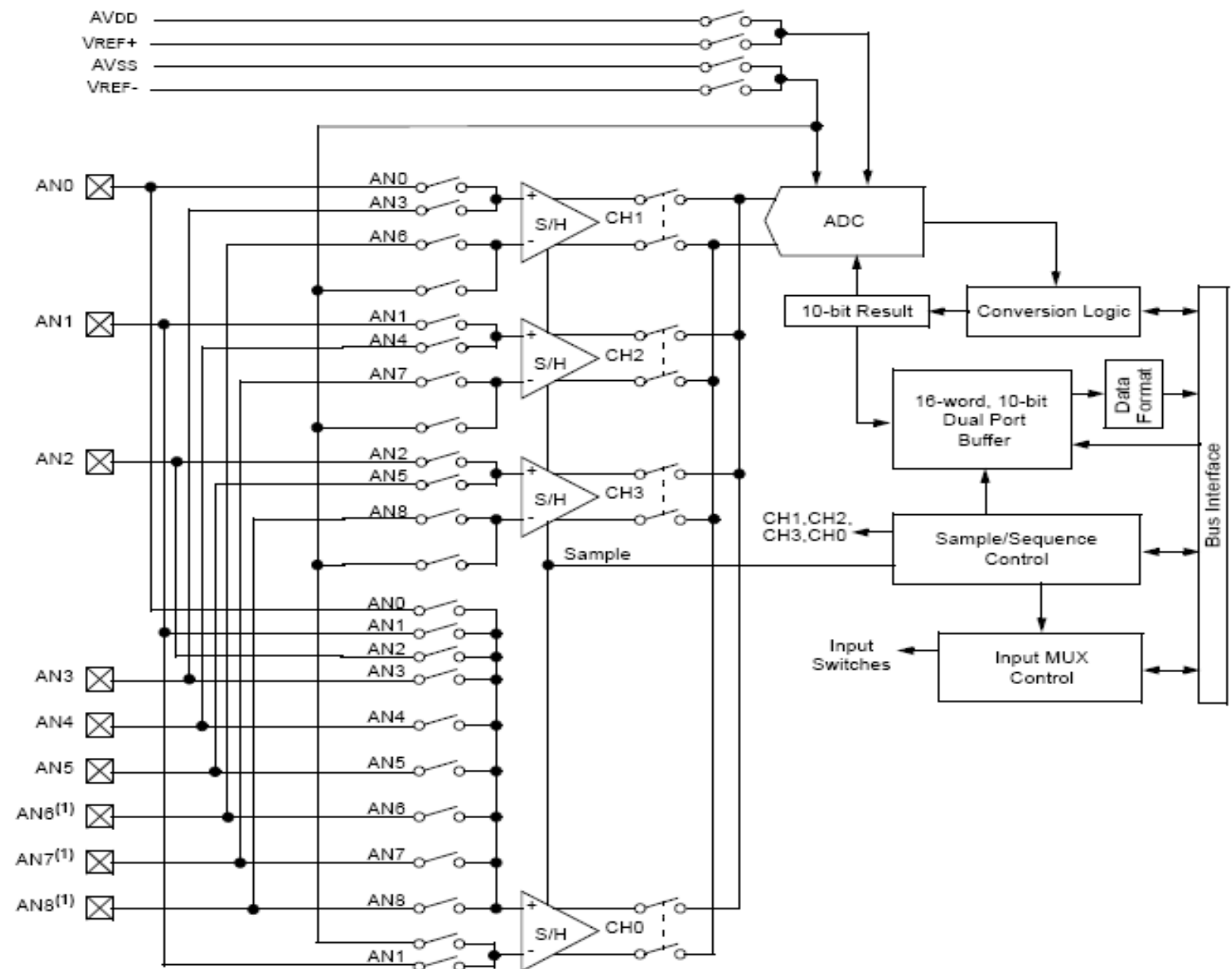
Os registradores ADCON1, ADCON2 e ADCON3 controlam a operação do módulo ADC. O registrador ADCHS seleciona os pinos de entrada a serem conectados aos amplificadores S/H. O registrador ADPCFG configura os pinos de entrada analógica como entradas analógicas ou como Entrada/Saída digital. O registrador ADCSSL seleciona as entradas a serem seqüencialmente escaneadas.

## **BUFFER RESULTADO DO**

### **ADC**

- ❑ O módulo contém duas portas RAM de 16 palavras, chamado de ADCBUF, para bufferar o resultado do ADC. As 16 locações de buffers são referidas como ADCBUF0, ADCBUF1, ... , ADCBUFE, ADCBUFF. O buffer resultado é um buffer de só leitura.

## **10-BIT, HIGH-SPEED ADC FUNCTIONAL BLOCK DIAGRAM**



Note 1: Not available on dsPIC30F4012 devices.

ADCON1: A/D Control Register 1					
Bit	Nome	Descrição	Bit	Nome	Descrição
0	DONE	<b>A/D Conversion Status bit (Rev. B silicon or later)</b> 1 = A/D conversion is done 0 = A/D conversion is NOT done Cleared by software or start of a new conversion. Clearing this bit will not effect any operation in progress.	9-8	FORM<1:0>	<b>Data Output Format bits</b> 11 = Signed Fractional (DOUT = sddd dddd dd00 0000) 10 = Fractional (DOUT = dddd dddd dd00 0000) 01 = Signed Integer (DOUT = ssss sssd dddd dddd) 00 = Integer (DOUT = 0000 00dd dddd dddd)
1	SAMP	<b>A/D Sample Enable bit</b> 1 = At least one A/D sample/hold amplifier is sampling 0 = A/D sample/hold amplifiers are holding When ASAM = 0, writing '1' to this bit will start sampling When SSRC = 000, writing '0' to this bit will end sampling and start conversion	12-10	-	<b>Não Implementado. Ler como 0</b>
2	ASAM	<b>A/D Sample Auto-Start bit</b> 1 = Sampling begins immediately after last conversion completes. SAMP bit is auto set 0 = Sampling begins when SAMP bit set	13	ADSIDL	<b>Stop in Idle Mode bit</b> 1 = Discontinue module operation when device enters Idle mode 0 = Continue module operation in Idle mode
3	SIMSAM	<b>Simultaneous Sample Select bit (only applicable when CHPS = 01 or 1x)</b> 1 = Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS = 1x) or Samples CH0 and CH1 simultaneously (when CHPS = 01) 0 = Samples multiple channels individually in sequence	14	-	<b>Não Implementado. Ler como 0</b>
4	-	<b>Não Implementado. Ler como 0</b>	15	ADON	<b>A/D Operating Mode bit</b> 1 = A/D converter module is operating 0 = A/D converter is off
7-5	SSRC<2:0>	<b>Conversion Trigger Source Select bits</b> 111 = Internal counter ends sampling and starts conversion (auto convert) 110 = Reserved 101 = Reserved 100 = Reserved 011 = Motor Control PWM interval ends sampling and starts conversion 010 = GP Timer3 compare ends sampling and starts conversion 001 = Active transition on INT0 pin ends sampling and starts conversion			
		000=Clearing SAMP bit ends sampling and starts conversion			

ADCON2: A/D Control Register 2																							
Bit	Nome	Descrição	Bit	Nome	Descrição																		
0	ALTS	<b>Alternate Input Sample Mode Select bit</b> 1 = Uses MUX A input multiplexer settings for first sample, then alternate between MUX B and MUX A input multiplexer settings for all subsequent samples 0 = Always use MUX A input multiplexer settings	9-8	CHPS<1:0>:	<b>Selects Channels Utilized bits</b> 1x = Converts CH0, CH1, CH2 and CH3 01 = Converts CH0 and CH1 00 = Converts CH0 When SIMSAM bit (ADCON1<3>) = 0 multiple channels sampled sequentially When SMSAM bit (ADCON1<3>) = 1 multiple channels sampled as in CHPS<1:0>																		
1	BUFM	<b>Buffer Mode Select bit</b> 1 = Buffer configured as two 8-word buffers ADCBUF(15...8), ADCBUF(7...0) 0 = Buffer configured as one 16-word buffer ADCBUF(15...0)	10	CSCNA	<b>Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit</b> 1 = Scan inputs 0 = Do not scan inputs																		
5-2	SMPI<3:0>	<b>Sample/Convert Sequences Per Interrupt Selection bits</b> 1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence 1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence ..... 0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence 0000 = Interrupts at the completion of conversion for each sample/convert sequence	11	-	<b>Não Implementado. Ler como 0</b>																		
6	-	<b>Não Implementado. Ler como 0</b>	12	Reservado	<b>Usuário escreverá ‘0’ nesta locação</b>																		
7	BUFS	<b>Buffer Fill Status bit</b> Only valid when BUFM = 1 (ADRES split into 2 x 8-word buffers). 1 = A/D is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7 0 = A/D is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF	15-13	VCFG<2:0>	<b>Voltage Reference Configuration bits</b> <table><tr><th></th><th>A/D VREFH</th><th>A/D VREFL</th></tr><tr><td>000</td><td>AVDD</td><td>AVSS</td></tr><tr><td>001</td><td>External VREF+ pin</td><td>AVSS</td></tr><tr><td>010</td><td>AVDD</td><td>External VREF- pin</td></tr><tr><td>011</td><td>External VREF+ pin</td><td>External VREF- pin</td></tr><tr><td>1XX</td><td>AVDD</td><td>AVSS</td></tr></table>		A/D VREFH	A/D VREFL	000	AVDD	AVSS	001	External VREF+ pin	AVSS	010	AVDD	External VREF- pin	011	External VREF+ pin	External VREF- pin	1XX	AVDD	AVSS
	A/D VREFH	A/D VREFL																					
000	AVDD	AVSS																					
001	External VREF+ pin	AVSS																					
010	AVDD	External VREF- pin																					
011	External VREF+ pin	External VREF- pin																					
1XX	AVDD	AVSS																					

ADCON3: A/D Control Register 3					
Bit	Nome	Descrição	Bit	Nome	Descrição
5-0	ADCS<5:0>	<b>A/D Conversion Clock Select bits</b> 11111 = TCY/2 • (ADCS<5:0> + 1) = 32 • TCY ..... 000001 = TCY/2 • (ADCS<5:0> + 1) = TCY 000000 = TCY/2 • (ADCS<5:0> + 1) = TCY/2	12-8	SAMC<4:0>	<b>Auto-Sample Time bits</b> 11111 = 31 TAD ..... 00001 = 1 TAD 00000 = 0 TAD (only allowed if performing sequential conversions using more than one S/H amplifier)
6	-	<b>Não Implementado. Ler como 0</b>	15-13	-	<b>Não Implementado. Ler como 0</b>
7	ADRC	<b>A/D Conversion Clock Source bit</b> 1 = A/D internal RC clock 0 = Clock derived from system clock			

ADCHS: A/D Input Channel Select Register					
Bit	Nome	Descrição	Bit	Nome	Descrição
3-0	CH0SA<3:0>	<b>Channel 0 Positive Input Select for MUX A Multiplexer Setting bits</b> 1111 = Channel 0 positive input is AN15 1110 = Channel 0 positive input is AN14 1101 = Channel 0 positive input is AN13 • • • 0001 = Channel 0 positive input is AN1 0000 = Channel 0 positive input is AN0	11-8	CH0SB<3:0>	<b>Channel 0 Positive Input Select for MUX B Multiplexer Setting bits</b> Same definition as bits 3-0 ( <b>Note</b> )  <b>Note:</b> The analog input multiplexer supports two input setting configurations, denoted MUX A and MUX B. ADCHS<15:8> determine the settings for MUX B, and ADCHS<7:0> determine the settings for MUX A. Both sets of control bits function identically.
4	CH0NA	<b>Channel 0 Negative Input Select for MUX A Multiplexer Setting bit</b> 1 = Channel 0 negative input is AN1 0 = Channel 0 negative input is VREFbit	12	CH0NB	<b>Channel 0 Negative Input Select for MUX B Multiplexer Setting bit</b> Same definition as bit 4 ( <b>Note</b> )
5	CH123SA	<b>Channel 1, 2, 3 Positive Input Select for MUX A Multiplexer Setting bit</b> 1 = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5 0 = CH1 positive input is AN0, CH2 positive input is AN1, CH3 positive input is AN2	13	CH123SB	<b>Channel 1, 2, 3 Positive Input Select for MUX B Multiplexer Setting bit</b> Same definition as bit 5 ( <b>Note</b> )
7-6	CH123NA<1:0>	<b>Channel 1, 2, 3 Negative Input Select for MUX A Multiplexer Setting bits</b> 11 = CH1 negative input is AN9, CH2 negative input is AN10, CH3 negative input is AN11 10 = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8 0x = CH1, CH2, CH3 negative input is VREFbit	15-14	CH123NB<1:0>	<b>Channel 1, 2, 3 Negative Input Select for MUX B Multiplexer Setting bits</b> Same definition as bits 6-7 ( <b>Note</b> )

ADPCFG: A/D Port Configuration Register		
Bit	Nome	Descrição
15-0	PCFG<15:0>	<b>Analog Input Pin Configuration Control bits</b> 1 = Analog input pin in Digital mode, port read input enabled, A/D input multiplexer input connected to AVSS 0 = Analog input pin in Analog mode, port read input disabled, A/D samples pin voltage

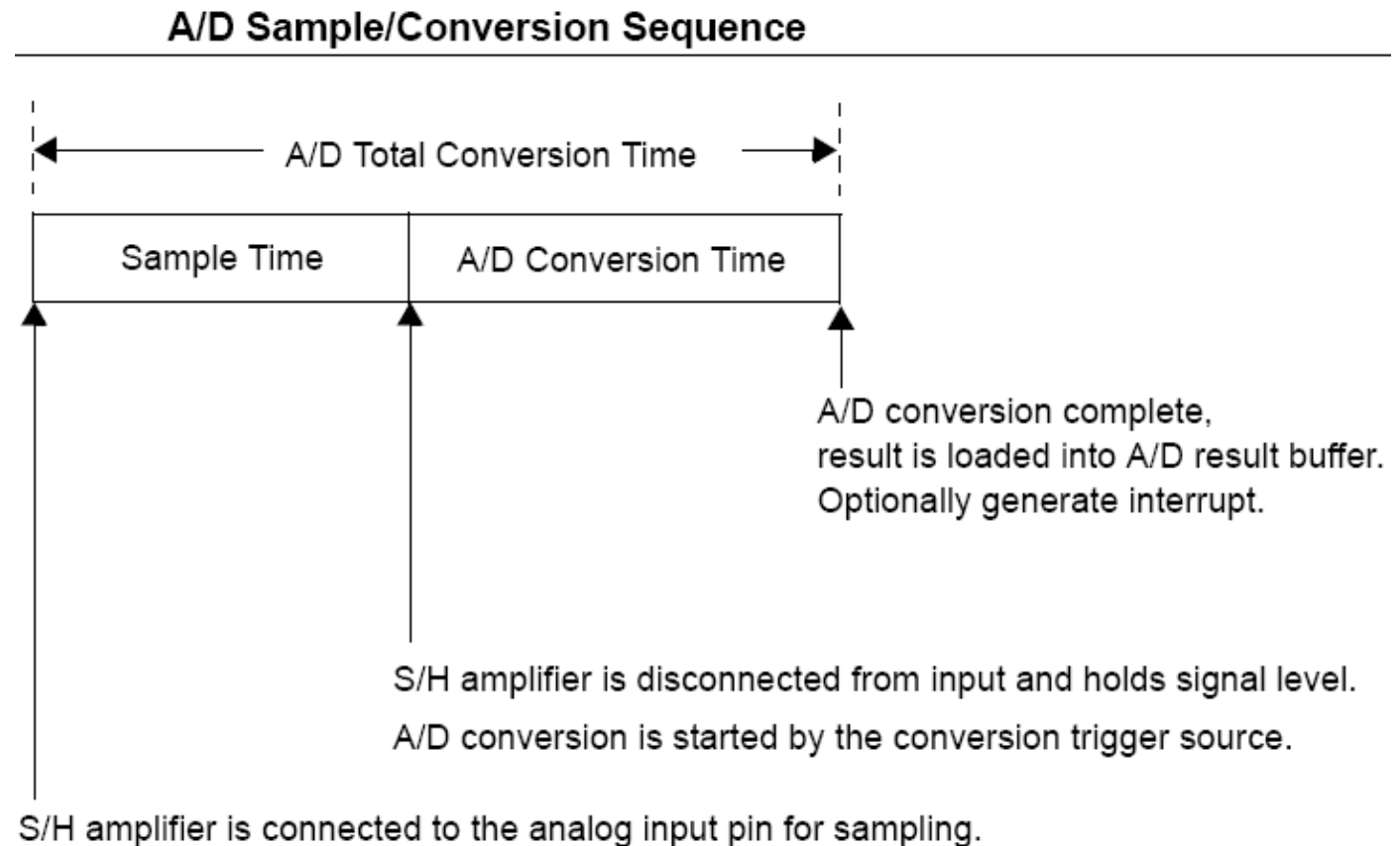
  

ADCSSL: A/D Input Scan Select Register		
Bit	Nome	Descrição
15-0	CSSL <15:0>	<b>A/D Input Pin Scan Selection bits</b> 1 = Select ANx for input scan 0 = Skip ANx for input scan

## Terminologia e Sequência de

### Conversão

- ❑ Na figura mostra-se uma sequência de conversão e os termos que são usados. Uma amostragem da tensão no pino de entrada analógico é executado pelo amplificador S/H. O conversor A/D de 10 bits tem 4 canais S/H em total, designados como CH0,...,CH3. Os canais S/H são conectados aos pinos de entrada analógica via o multiplexador de entrada analógica.

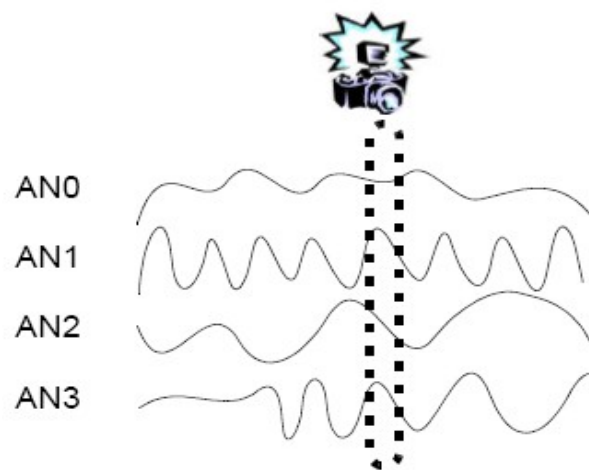


**Terminologia e seqüência de conversão (continuação...)**

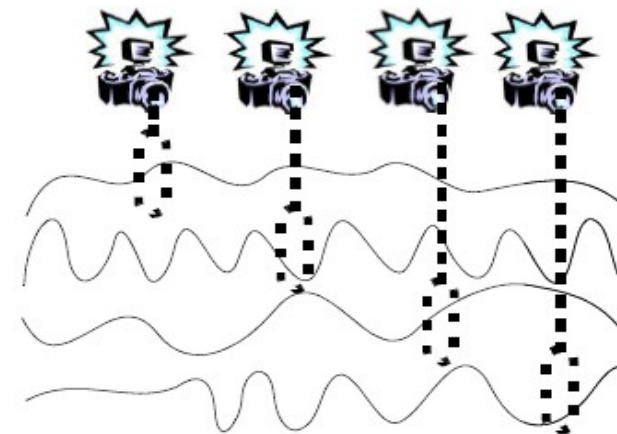
- ❑ O multiplexador de entrada analógico é controlado pelo registrador ADCHS. Existem dois conjuntos de bits para controle do multiplexador no registrador ADCHS que funcionam identicamente. Estes dois conjuntos de bits de controle permitem programar duas configurações diferentes do multiplexador de entrada analógico, os quais são chamados de MUX A e MUX B. O conversor A/D pode opcionalmente chavear entre MUX A e MUX B entre conversões e também pode escanear através de uma série de entradas analógicas.
- ❑ Tempo de amostragem é o tempo que o amplificador S/H do modulo A/D é conectado ao pino de entrada analógico. Este tempo pode começar manualmente setando o bit SAMP (ADCON1<1>) ou começar automaticamente pelo hardware do conversor A/D. Este tempo de amostragem é finalizado manualmente zerando o bit de controle SAMP pelo software do usuário ou automaticamente pela fonte de disparo do conversor.
- ❑ Tempo de conversão é o tempo requerido pelo conversor A/D para converter a tensão mantida pelo amplificador S/H. O A/D é desconectado do pino de entrada analógico ao final do tempo de amostragem. O conversor A/D requer um ciclo de clock A/D (TAD) para converter cada bit do resultado mais um ciclo de clock adicional. Um total de 12 TADs são requeridos para executar uma conversão completa. Quando o tempo de conversão é completa, o resultado é carregado em um dos 16 registradores resultado do A/D (ADCBUF0,..., ADCBUFF), o S/H pode ser reconectada ao pino de entrada e a interrupção da CPU pode ser gerada.

## Terminologia e seqüência de conversão (continuação...)

- ❑ A soma dos tempos de amostragem e de conversão A/D fornece o tempo de conversão total. Existe um tempo mínimo de amostragem para assegurar que o amplificador S/H dará a precisão desejada para a conversão A/D. O usuário deve selecionar uma opção de entrada de clock que não viole o mínimo TAD especificado.
- ❑ O conversor permite muitas opções para especificar a seqüência amostragem/conversão. Esta seqüência pode ser muito simples, como se mostra na figura, onde se usa um amplificador S/H e outra com quatro amplificadores S/H para fazer quatro conversões. O número de amplificadores S/H, ou canais por amostra, usado na seqüência amostra/conversor é determinado pelos bits de controle CHPS e quando se usa vários canais S/H simultaneamente ou sequencialmente este é controlado pelo bit SIMSAM (ADCON1<3>).



Simultaneous  
Sampling



Sequential  
Sampling



**Terminologia e seqüência de conversão (continuação...)**

- ❑ O tempo de início de amostragem pode ser controlado pelo bit SAMP ou automaticamente pelo hardware. Quando o conversor A/D opera no modo Auto-Sample, o amplificador S/H é reconectado ao pino de entrada analógica ao final da conversão na sequência amostragem/conversão. A função Auto-Sample é controlada pelo bit ASAM (ADCON1<2>).
- ❑ A fonte de disparo da conversão finaliza o tempo de amostragem e começa uma conversão A/D ou uma sequência amostragem/conversão, esta é selecionada pelo bit de controle SSRC. O disparo da conversão pode ser tomada de uma variedade de fontes de hardware, ou pode ser controlada manualmente em software zerando o bit de controle SAMP. Uma das fontes de disparo da conversão é uma auto-conversão. O tempo entre auto-conversões é setada por um contador e o clock do A/D.
- ❑ O modo Auto-sample e disparo de auto-conversão podem ser usados juntos para fornecer conversões automáticas sem fim e sem intervenção do software. Uma interrupção pode ser gerada ao final de cada sequência ou múltiplas sequências de amostragem/conversão como determinado pelo valor dos bits de controle SMPI (ADCON2<5:2>). O número de sequências entre interrupções pode variar entre 1 e 16. O usuário notará que o buffer de conversão A/D mantém 16 resultados quando o valor do SMPI é selecionado. O número total de conversões resultante entre interrupções é o produto dos canais por amostra e o valor do SMPI. O número total de conversões entre interrupções não excederá o comprimento do buffer.

## **Configuração do módulo ADC**

### 0) Configurar o módulo A/D

- ★ Selecionar os pinos da porta como entradas analógicas ADPCFG<15:0>.
- ★ Selecionar a tensão de referência para acertar a faixa das entradas analógicas ADCON2 <15:13>
- ★ Selecionar o clock de conversão analógica para acertar a relação da data desejada com clock do processador ADCON3<5:0>
- ★ Determinar quantos canais S/H serão usados ADCON2<9:8> e ADPCFG<15:0>
- ★ Determinar quantas amostragens acontecerão ADCON1<3> e ADCSSL <15:0>
- ★ Determinar quantas entradas serão alocadas para os canais S/H, ADCHS<15:0>
- ★ Selecionar a apropriada seqüência amostragem/conversão ADCON1<7:0> e ADCON3<12:8>
- ★ Selecionar quantos resultados de conversão serão apresentados no buffer ADCON1<9:8>
- ★ Selecionar a relação de interrupção ADCON2<5:9>
- ★ Ligar o módulo A/D ADCON1<15>.

- 1) Configurar a interrupção A/D (se requerido): zerar o bit ADIF e selecionar a prioridade da interrupção.
- 2) Iniciar a amostragem.
- 3) Esperar o tempo de aquisição requerido.
- 4) Finalizar disparo da aquisição, começar conversão.
- 5) Esperar para completar a conversão A/D: Esperando pela interrupção A/D ou esperando pelo bit DONE ser setado.
- 6) Ler o buffer resultado A/D, zerar ADIF se requerido.

## Selecionando a fonte da tensão de referência

- ❑ As referências de tensão para conversões A/D são selecionadas usando os bits de controle VCFG<2:0> (ADCON2<15:13>). A referência de tensão superior (VREFH) e a tensão de referência baixa (VREFL) podem ser as tensões internas AVDD e AVSS ou os pinos de entrada VREF+ e VREF-.
- ❑ Os pinos de referência de tensão externa podem ser compartilhados com as entradas AN0 e AN1. O conversor A/D pode executar conversões nestes pinos quando eles são compartilhados com os pinos de entrada VREF+ e VREF-. As tensões aplicadas aos pinos de referência externa devem reunir certas características que devem ser vistas no datasheet.
- ❑ VREF+ e VREF- externo deve ser selecionado para conversões até de 500 Ksps.

dsPIC30F 10-bit A/D Converter Conversion Rates						
A/D Speed	TAD Minimum	Sampling Time Min	R <sub>s</sub> Max	VDD	Temperature	A/D Channels Configuration
Up to 1 Msps <sup>(1)</sup>	83.33 ns	12 TAD	500 Ω	4.5V to 5.5V	-40°C to +85°C	
Up to 750 ksps <sup>(1)</sup>	95.24 ns	2 TAD	500 Ω	4.5V to 5.5V	-40°C to +85°C	
Up to 600 ksps <sup>(1)</sup>	138.89 ns	12 TAD	500 Ω	3.0V to 5.5V	-40°C to +125°C	
Up to 500 ksps	153.85 ns	1 TAD	5.0 kΩ	4.5V to 5.5V	-40°C to +125°C	
Up to 300 ksps	256.41 ns	1 TAD	5.0 kΩ	3.0V to 5.5V	-40°C to +125°C	

**Selecionando o clock de Conversão do A/D**

- ❑ O conversor A/D tem uma relação máxima na qual as conversões serão completadas. Um clock do modulo analógico, TAD, controla a temporização da conversão. A conversão A/D requer de 12 períodos de clock (12 TAD). O clock A/D é derivado desde o clock de instrução (TCY) do dispositivo ou do clock RC interno. O período do clock de conversão A/D é selecionado por software usando um contador de seis bits. Existem 64 possíveis opções para o TAD, especificado pelos bits ADCS<5:0> (ADCON3<5:0>). A seguinte equação dá o valor a TAD como uma função dos bits de controle ADCS e o período do clock de instrução, TCY.

$$T_{AD} = \frac{TCY(ADCS + 1)}{2}$$

$$ADCS = \frac{2 T_{AD}}{TCY} - 1$$

- ❑ Para conversões A/D corretas, o clock de conversão A/D (TAD) deve ser selecionado para fornecer um tempo mínimo TAD de 83,33 ns (para conversões de 1Msps).
- ❑ O conversor A/D tem uma fonte de clock interno RC dedicado que pode ser usado para executar conversões. O clock interno RC poderia ser usado quando conversões A/D são executadas no modo Sleep. O oscilador interno RC é selecionado setando o bit ADRC (ADCON3<7>). Quando o bit ADRC é setado, os bits ADCS<5:0> não tem efeito sobre a operação A/D.
- ❑ No seguinte exemplo mostramos o cálculo para os bits ADCS<5:0>, assumindo um dispositivo operando a velocidade de 30 MIPS (para conversões de 500 Hz).

#### A/D CONVERSION CLOCK CALCULATION

$$TAD = 154 \text{ nsec}$$

$$TcY = 33 \text{ nsec (30 MIPS)}$$

$$\begin{aligned} ADCS<5:0> &= 2 \frac{TAD}{TcY} - 1 \\ &= 2 \cdot \frac{154 \text{ nsec}}{33 \text{ nsec}} - 1 \\ &= 8.33 \end{aligned}$$

Therefore,

Set ADCS<5:0> = 9

$$\begin{aligned} \text{Actual TAD} &= \frac{TcY}{2} (ADCS<5:0> + 1) \\ &= \frac{33 \text{ nsec}}{2} (9 + 1) \\ &= 165 \text{ nsec} \end{aligned}$$

### **Programando o começo do disparo da conversão**

- ❑ O disparo da conversão termina a aquisição e começa a conversão requerida. Os bits SSRC<2:0> selecionam a fonte de disparo da conversão, estes fornecem até cinco fontes alternativas de disparo de conversão. Quando SSRC<2:0> = 000, o disparo da conversão está sob o controle do software. Zerando o bit SAMP causa o disparo da conversão. Quando SSRC<2:0> = 111 (modo Auto-Start), o disparo da conversão está baixo o controle do clock A/D. Os bits SAMC selecionam o número de clocks A/D entre o começo da aquisição e o começo da conversão. Este fornece a relação de conversão mais veloz sobre múltiplos canais. SAMC sempre deve estar a no mínimo um ciclo de clock. Outras fontes de disparo podem vir desde módulos do timer, PWM controle do motor ou interrupções externas.
- ❑ Para especificar o ADC à velocidade de conversão máxima especificada, a opção de disparo de auto-conversão será selecionada (SSRC = 111) e os bits de tempo de auto-sample serão setados a '1' TAD (SAMC = 00001). Esta configuração dá um período de conversão total (sample + convert) de 13 TAD. O uso de algum outro disparo de conversão resulta em adicionais ciclos de TAD para sincronizar o evento externo ao ADC.

**Abortando uma conversão**

- ❑ Zerando o bit ADON durante uma conversão aborta a conversão atual e pára a sequência de amostragem. O ADCBUFx não é atualizado, isto é, o ADCBUFx continuará contendo o valor da última conversão completa. Se quando se zera o bit ADON coincide com um auto-start, o zero do bit tem maior prioridade. Depois que a conversão é abortada, é requerido 2 TAD de espera antes que a próxima amostragem possa ser iniciada setando o bit SAMP. Se a amostragem sequencial é especificada, o A/D continua ao próximo pulso de amostragem, o qual corresponde com o próximo canal convertido. Se amostragens simultâneas são especificadas, o A/D continua com a sequência de conversão do próximo grupo multicanal.

**Lendo o buffer resultado do A/D**

- ❑ A RAM é de 10 bits de largura, mas o dado é automaticamente formatado para um dos quatro formatos selecionáveis. Os bits FORM<1:0> (ADCON1<9:8>) seleciona o formato. A formatação por hardware fornece um resultado de 16 bits sobre o barramento de dados para todos os formatos de dados.

## A/D Output Data Formats

RAM Contents:

d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Read to Bus:

Integer

0	0	0	0	0	0	d09	d08	d07	d06	d05	d04	d03	d02	d01	d00
---	---	---	---	---	---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Signed Integer

$\overline{\text{d09}}$	$\overline{\text{d09}}$	$\overline{\text{d09}}$	$\overline{\text{d09}}$	$\overline{\text{d09}}$	$\overline{\text{d09}}$	$\overline{\text{d09}}$	d08	d07	d06	d05	d04	d03	d02	d01	d00
-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----

Fractional (1.15)

d09	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	---	---	---	---	---

Signed Fractional (1.15)

$\overline{\text{d09}}$	d08	d07	d06	d05	d04	d03	d02	d01	d00	0	0	0	0	0	0
-------------------------	-----	-----	-----	-----	-----	-----	-----	-----	-----	---	---	---	---	---	---



### Numerical Equivalents of Various Result Codes

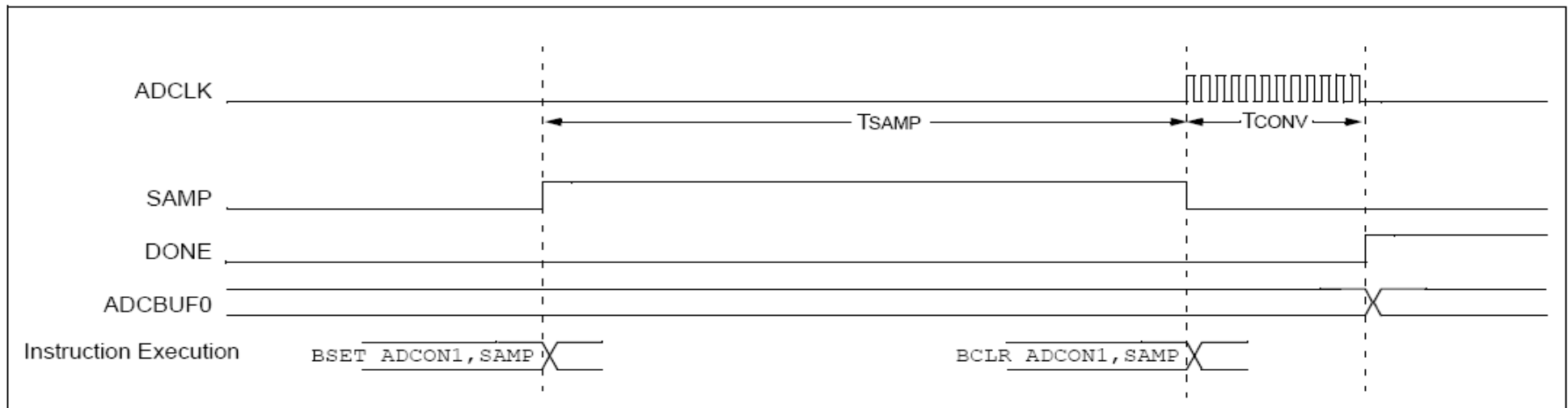
V <sub>IN</sub> /V <sub>REF</sub>	10-bit Output Code	16-bit Integer Format	16-bit Signed Integer Format	16-bit Fractional Format	16-bit Signed Fractional Format
1023/1024	11 1111 1111	0000 0011 1111 1111 = 1023	0000 0001 1111 1111 = 511	1111 1111 1100 0000 = 0.999	0111 1111 1100 0000 = 0.499
1022/1024	11 1111 1110	0000 0011 1111 1110 = 1022	0000 0001 1111 1110 = 510	1111 1111 1000 0000 = 0.998	0111 1111 1000 0000 = 0.498
...					
513/1024	10 0000 0001	0000 0010 0000 0001 = 513	0000 0000 0000 0001 = 1	1000 0000 0100 0000 = 0.501	0 000 0000 0100 0000 = 0.001
512/1024	10 0000 0000	0000 0010 0000 0000 = 512	0000 0000 0000 0000 = 0	1000 0000 0000 0000 = 0.500	0000 0000 0000 0000 = 0.000
511/1024	01 1111 1111	0000 0001 1111 1111 = 511	1111 1111 1111 1111 = -1	0111 1111 1100 0000 = .499	1111 1111 1100 0000 = -0.001
...					
1/1024	00 0000 0001	0000 0000 0000 0001 = 1	1111 1110 0000 0001 = -511	0000 0000 0100 0000 = 0.001	1000 0000 0100 0000 = -0.499
0/1024	00 0000 0000	0000 0000 0000 0000 = 0	1111 1110 0000 0000 = -512	0000 0000 0000 0000 = 0.000	1000 0000 0000 0000 = -0.500

### **Escaneando várias entradas com o Canal 0**

- ❑ O canal 0 tem a habilidade de escanear através de um vetor de seleção de entradas. O bit CSCNA (ADCON2<10>) habilita as entradas do canal 0 (CH0) para serem escaneadas através de um número selecionado de entradas analógicas. Quando CSCNA é setada ("1"), os bits CH0SA<3:0> são ignorados. O registrador ADCSSL especifica as entradas a serem escaneadas. Cada bit no registrador ADCSSL corresponde a uma entrada analógica. Bit 0 corresponde a AN0, bit 1 corresponde a AN1 e assim por diante. Se um bit no registrador ADCSSL é "1", a entrada correspondente é parte da sequência de escaneamento. As entradas são sempre escaneadas desde as entradas numeradas mais baixas para as superiores, começando no primeiro canal selecionado depois que acontece uma interrupção. Os bits ADCSSL somente especificam a entrada positiva do canal. O bit CH0NA seleciona a entrada negativa do canal durante o escaneamento.
- ❑ Se o bit ALTS é "1", o escaneamento só se aplica à seleção das entradas do MUX A. A seleção da entrada do MUX B, como especificado pelo CH0SB<3:0>, selecionará alternadamente a entrada de canal 0 entre um conjunto de entradas escaneadas especificado pelo registrador ADCSSL e uma entrada fixa especificada pelos bits CH0SB.

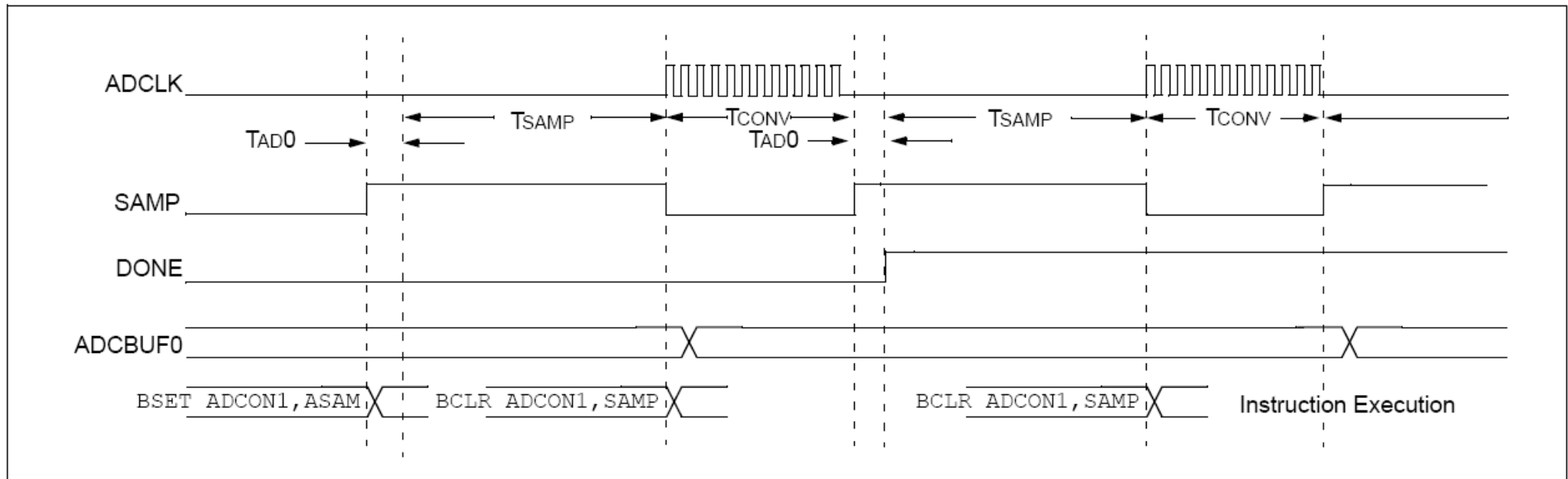
## **Como começar a Amostragem**

- ❑ **Manual** : setando o bit SAMP (ADCON1<1>) causa que o A/D comece a amostragem. Uma de várias opções podem ser usadas para finalizar a amostragem e completar a conversão. A amostragem não finaliza até que o bit SAMP é zerado.



- ❑ **Automático** : setando o bit ASAM (ADCON1<2>) causa que o A/D comece automaticamente a amostragem de um canal sempre que uma conversão no esteja ativa no canal. Uma de várias opções podem ser usadas para finalizar a amostragem e completar a conversão. Se o bit SIMSAM especifica amostragem sequencial, é retomada a amostragem de um canal depois que a conversão do canal foi completada. Se o bit SIMSAM especifica amostragem simultânea, a amostragem de um canal é retomada depois que a conversão de todos os canais tenham sido completadas.

- ❑ O bit ASAM não pode ser modificado enquanto o conversor A/D está ligado. O módulo A/D toma algum tempo para se estabilizar, portanto, se a amostragem automática é habilitada, não há garantia que o primeiro resultado do A/D seja correto até que o módulo se estabilize. Poderia ser necessário descartar o primeiro resultado do A/D dependendo da velocidade do clock.



```

/* Conversão MANUAL A/D com um potenciometro ligado a RB8. Os valores serão mostrados no LCD. O tempo de amostragem será definido pelo
usuário. O tempo de conversão será de 12 TAD para cada entrada analógica. */

#include <p30f4011.h> //arquivos de definições do dsPIC30f4011

// Função para trabalhar com o LCD

//Configurações do dsPIC
_FOSC(CSW_FSCM_OFF & XT_PLL8); //Cristal de 8MHz
_FWDT(WDT_OFF); //Watchdog desligado
_FBORPOR(PBOR_OFF & MCLR_EN); //Brown Out desligado, Master clear habilitado
_FGS(CODE_PROT_OFF); //Código sem proteção
// ----- PROGRAMA PRINCIPAL-----//
int main(void)
{
//----- CONFIGURAÇÃO DE ENTRADA/SAÍDA -----//
    ADPCFG = 0xFEFF; // Pino RB8 como entrada analógica (0 = analógica, 1=digital)
    TRISB = 0xFFFF; // Porta B como entrada
//----- CONFIGURAÇÃO DO CONVERSOR A/D -----//
//Configuração do SFR ADCON1:Adc desligado, Formato de saída: inteiro, Amostragem/conversão manual
    ADCON1=0;
//Configuração do SRF ADCON2: tensão de referência: AVDD e AVSS, sem varredura, conversão pelo canal 0, interrupção após uma
//amostra buffer como palavra de 16 bits
    ADCON2=0;
//Configuração do SRF ADCON3: para o clock do ADC de 500kHz, tempo de auto-amostragem = 0 TAD, fonte do clock: ciclo da máquina
    ADCON3=0x0007; //Tad= 4 * Tcy = 4*62,5 ns = 250 ns > 153,85 ns (quando Vdd = 4,5 a 5,5V)
//
    ADCON3 = 0x000B; // Tad = 6 * Tcy = 6* 62,5ns = 375ns >256,41ns (quando Vdd = 3 a 5,5V);
//Configuração do SRF ADCHS: seleciona o canal CH0, configura entrada analógica AN8 (RB8), entrada de referência negativa do
//CHO igual a Vref-
    ADCHS=0x0000;
    ADCHSbits.CH0SA = 8; // seleciona a entrada analogica 8
//Configuração do SRF ADCSSL: varredura desativada
    ADCSSL=0;
    ADCON1bits.ADON = 1; //Ativa o ADC
// ----- CONFIGURAÇÃO DO LCD -----//
//
    Inicializa_LCD
    float tensao;
    int convertido;
    while(1)
    {
        ADCON1bits.SAMP=1; //Inicia a amostragem
        delay_us(10); //Espera a amostragem
        ADCON1bits.SAMP=0; //Para a amostragem e inicia a conversão
        while(!ADCON1bits.DONE); //Aguarda o fim da conversão
        convertido = ADCBUF0; //Ler o valor convertido no canal 0 do ADC
        tensao = (convertido*5.0)/1023; // Cálculo da tensão de entrada
        limpa_LCD(); // Limpa o LCD
        variavel_LCD(tensao); // Mostra valor da tensão no LCD
        delay_ms(1000);
    }
}

```

```

/* Conversao automatica A/D com um potenciometro ligado a RB8.Os valores de cada um serão mostrados no PC. O tempo de amostragem será
definido pelo usuário o tempo de conversão será de 12 TAD para cada entrada analógica.*/
#include <p30f4011.h> //arquivos de definições do dsPIC30f4011

// Função de inicialização do LCD

// ***** Função de atraso de tempo. Gera um atraso de 'tempo' us

//Configurações do dsPIC
_FOSC(CSW_FSCM_OFF & XT_PLL8); //Cristal de 8MHz
_FWDT(WDT_OFF); //Watchdog desligado
_FBORPOR(PBOR_OFF & MCLR_EN); //Brown Out desligado, Master clear habilitado
_FGS(CODE_PROT_OFF); //Código sem proteção
// ----- PROGRAMA PRINCIPAL-----//
int main(void)
{
//----- CONFIGURAÇÃO DE ENTRADA/SAÍDA -----//
ADPCFG = 0xFEFF; // Pino RB8 como entrada analógica (0 = analógica, 1=digital)
TRISB = 0xFFFF; // Porta B como entrada
//----- CONFIGURAÇÃO DO CONVERSOR A/D -----//
//Configuração do SFR ADCON1.Adc desligado. Formato de saída: inteiro. Amostragem/conversão automatica
    ADCON1 = 0X0004; // conversão automatica começa depois da amostragem
//Configuração do SRF ADCON2. tensão de referência: AVDD e AVSS. sem varredura. conversão pelo canal 0. interrupção após uma amostra
// buffer como palavra de 16 bits
    ADCON2=0;
//Configuração do SRF ADCON3: para o clock do ADC de 500kHz, tempo de amostragem = dado pelo usuario, fonte do clock: clock do sistema
Tcy, ADCS = 7, SAMC = 0 Tad
    ADCON3=0x0007; //Tad= 4 * Tcy = 4*62,5 ns = 250 ns > 153,85 ns (quando Vdd = 4,5 a 5,5V)
// ADCON3 = 0x000B; // Tad = 6 * Tcy = 6* 62,5ns = 375ns >256,41ns (quando Vdd = 3 a 5,5V);

//Configuração do SRF ADCHS.seleciona o canal CH0. configura entrada analógica AN8 (RB8). referência negativa do CHO igual a Vref-
    ADCHS=0x0000;
    ADCHSbits.CH0SA = 8; // seleciona a entrada analogica 8
//Configuração do SRF ADCSSL. varredura desativada
    ADCSSL=0;
//Ativa o ADC
    ADCON1bits.ADON = 1;

    inicializa_LCD();
//-----
    int convertido;
    float tensao;
    while(1)
    {
        __delay_us(10);
        ADCON1bits.SAMP=0; //Para a amostragem
        while(!ADCON1bits.DONE); //Aguarda o fim da conversão
        convertido = ADCBUF0; //Ler o valor convertido no canal 0 do ADC
        tensao = (convertido*5.0)/1023; // Cálculo da tensão de entrada
        limpa_LCD(); // Limpa o LCD
        variavel_LCD(tensao); // Mostra valor da tensão no LCD
        delay_ms(1000);
    }
}

```

```

/* Conversão auto conversão do A/D com um potenciometro ligado a RB8. Os valores de cada um serão mostrados no LCD. O tempo de amostragem será definido por configuracao. O tempo de conversão será de 12 TAD para cada entrada analógica. */
#include <p30f4011.h> //arquivos de definições do dsPIC30f4011
// biblioteca do LCD
//Configurações do dsPIC
_FOSC(CSW_FSCM_OFF & XT_PLL8); //Cristal de 8MHz
_FWDT(WDT_OFF); //Watchdog desligado
_FBORPOR(PBOR_OFF & MCLR_EN); //Brown Out desligado, Master clear habilitado
_FGS(CODE_PROT_OFF); //Código sem proteção
// ----- PROGRAMA PRINCIPAL -----//
int main(void)
{
//----- CONFIGURAÇÃO DE ENTRADA/SAÍDA -----//
ADPCFG = 0xFEFF; // Pino RB8 como entrada analógica (0 = analógica, 1=digital)
TRISB = 0xFFFF; // Porta B como entrada
//----- CONFIGURAÇÃO DO CONVERSOR A/D -----//
//Configuração do SFR ADCON1:Adc desligado, Formato de saída: inteiro, os Conversion Trigger Source Select bits (SSRC=111) determina o disparo da conversão.
ADCON1=0x00E0;

//Configuração do SRF ADCON2: tensão de referência: AVDD e AVSS, sem varredura, conversão pelo canal 0, interrupção após uma amostra
//buffer como palavra de 16 bits
ADCON2=0;

//Configuração do SRF ADCON3: para o clock do ADC = 500kHz, tempo de amostragem = dado pelo usuario, fonte do clock: clock do sistema
Tcy, ADCS = 7, SAMC = 1F = 31*Tad (duracao do tempo de amostragem)
ADCON3=0x1F07; //Tad= 4 * Tcy = 4*62,5 ns = 250 ns > 153,85 ns (quando Vdd = 4,5 a 5,5V)

//Configuração do SRF ADCHS: seleciona o canal CH0, configura entrada analógica AN8 (RB8), entrada de referência negativa do CHO
//igual a Vref-
ADCHS=0x0000;
ADCHSbits.CH0SA = 8; // seleciona a entrada analógica 8

//Configuração do SRF ADCSSL: varredura desativada
ADCSSL=0;
ADCON1bits.ADON = 1; //Ativa o ADC
// ----- CONFIGURAÇÃO DA SERIAL -----//
inicializa_LCD(); // Inicialização do display LCD
// Declaração de variáveis
int convertido;
float tensao; // Declaração de variáveis
while(1)
{
    ADCON1bits.SAMP = 1; // Inicia a amostragem
    while(!ADCON1bits.DONE); // Aguarda o fim da conversão DONE = 0;
    convertido = ADCBUF0; //Ler o valor convertido na entrada analógica do ADC
    tensao = (convertido*5.0)/1023; // Cálculo da tensão de entrada
    limpa_LCD(); // Limpa o LCD
    variavel_LCD(tensao); // Mostra valor da tensão no LCD
    delay_ms(500);
}
}

```

```

// Conversao A/D com dois potenciometros ligados a RB8 e RB7. Os valores de cada um serão mostrados no LCD
// O tempo de amostragem será definido pelo usuário. O tempo de conversão será de 12 TAD para cada entrada analógica.
#include <p30f4011.h> //arquivos de definições do dsPIC30f4011
// Função de inicialização do LCD
//Rotina de inicialização manual do conversor A/D
void init_ADC_man()          //masc = mascara para selecionar os pinos que serão analogicos
{
    //ADPCFG = masc; // seleciona pinos de entrada analógicos
    ADCON1 = 0; // controle de sequencia de conversão manual
    ADCSSL = 0; // não é requerida a varredura ou escaneamento
    ADCON2 = 0; // usa MUXA, AVdd e AVss são usados como Vref+/-
    ADCON3 = 0x0007; // Tad = 4 x Tcy = 4* 62,5ns = 250 ns > 153,85 ns (quando Vdd = 4,5 a 5,5V); SAMC = 0 (não é levado em consideração
    //quando é conversão manual)
    ADCON1bits.ADON = 1; // liga o ADC
} //init_ADC_man
//Rotina básica de leitura (amostragem manual) do conversor A/D
int ler_ADC_man( int canal)
{
    ADCHS = canal; // seleciona canal de entrada analógica
    ADCON1bits.SAMP = 1; // começa amostragem
    delay_us(10); //tempo de amostragem
    ADCON1bits.SAMP = 0; // começa a conversão
    while (!ADCON1bits.DONE); // espera que complete a conversão
    return ADCBUF0; // le o resultado da conversão.
} // ler_ADC_man
//Configurações do dsPIC
_FOSC(CSW_FSCM_OFF & XT_PLL8); //Cristal de 8MHz
_FWDT(WDT_OFF); //Watchdog desligado
_FBORPOR(PBOR_OFF & MCLR_EN); //Brown Out desligado, Master clear habilitado
_FGS(CODE_PROT_OFF); //Código sem proteção

// ----- PROGRAMA PRINCIPAL-----//
int main(void)
{
    ADPCFG = 0xFE7F; // Pino RB8 e RB7 como entrada analógica (0 = analógica, 1=digital)
    TRISB = 0xFFFF; // Porta B como entrada
    inicializa_LCD();
    init_ADC_man();
    int convertido;
    float Vin;

    while(1)
    {
        convertido = ler_ADC_man(7); //Ler o valor convertido na entrada analógica do ADC
        Vin = convertido * 0.004887;
        //
        //
        convertido = ler_ADC_man(8); //Ler o valor convertido na entrada analógica do ADC
        Vin = convertido * 0.004887;
        //
        //
        delay_ms(1000);
    }
}

```